



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



Publication number : **0 689 325 A2**

(12)

## EUROPEAN PATENT APPLICATION

(21) Application number : **95304260.3**

(51) Int. Cl.<sup>6</sup> : **H04L 29/06**

(22) Date of filing : **20.06.95**

(30) Priority : **22.06.94 GB 9412553**

(43) Date of publication of application :  
**27.12.95 Bulletin 95/52**

(84) Designated Contracting States :  
**DE ES FR GB IT**

(71) Applicant : **AT & T GLOBAL INFORMATION  
SOLUTIONS INTERNATIONAL INC.  
1700 South Patterson Boulevard  
Dayton, Ohio 45479 (US)**

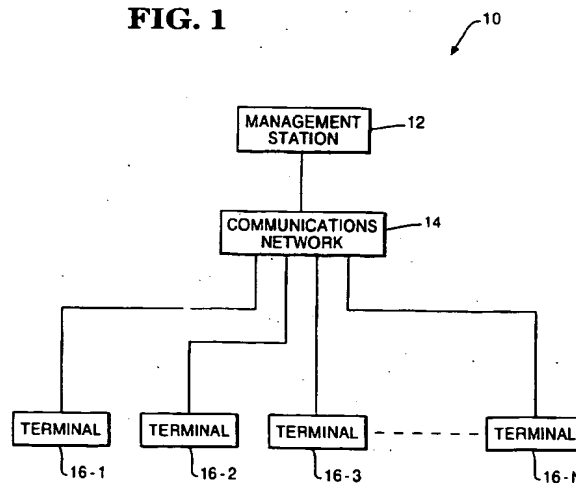
(72) Inventor : **Shepherd, Bruce  
5 McGill Road  
Carnoustie, Angus (GB)  
Inventor : McPherson, Robert P.  
4 Greystaine Road  
Invergowrie, Dundee (GB)**

(74) Representative : **Robinson, Robert George  
International Patent Department,  
AT&T GIS Limited,  
915 High Road,  
North Finchley  
London N12 8QJ (GB)**

### (54) Method and apparatus for distributing software

(57) In a software management system, a software package held in a management station (12) is transferred across a communications network (14) to a plurality of destination terminals (16), of which there may be a large number. In order to save on distribution time, the management station (12) can cause terminals (16) which have received the software package to transmit it to further terminals, until all terminals (16) have received the package.

FIG. 1



EP 0 689 325 A2

This invention relates to a method and apparatus for distributing software.

The invention has a particular application to the distribution of a software package to a plurality of terminals.

One example of a system wherein a plurality of terminals are connected to a management station is where the terminals are self-service terminals (SSTs) such as automated teller machines (ATMs). The terminals may be located at widely separated geographical locations, and are interconnected by a communications network such as a public data network, for example a network conforming to the CCITT X.25 standard. The transfer of signals over long distances via such a network may be via low speed lines and hence lengthy times may be involved in such transfer. With such a network of terminals, it is sometimes desirable that a common software package should be distributed to all the terminals in the network. Such software package may, for example, modify the manner in which the terminals operate and it is desirable that all the terminals should receive the common software package without undue delays.

It is an object of the present invention to provide a method and apparatus whereby software may be distributed to a plurality of terminals at high speed.

Therefore, according to one aspect of the present invention, there is provided a method of distributing a software package from a source station to a plurality of terminals, characterized by the steps of: transmitting said software package from said source station to at least one of said terminals; and utilizing at least one of said terminals which has received said software package to transmit said software package to another of said terminals.

According to another aspect of the present invention, there is provided a terminal system including a management station, a plurality of terminals and connection means adapted to interconnect said management station and said terminals, characterized in that said management station is adapted to transmit a software package to at least one of said terminals, and is that said management station is adapted to cause a terminal which has received said software package to transmit said software package to another of said terminals.

It will be appreciated that a method and apparatus according to the invention enable a significant saving in distribution time to be achieved by utilizing terminals which have received the software package to distribute the software package to other terminals.

One embodiment of the present invention will now be described by way of example, with reference to the accompanying drawings, in which:-

Fig. 1 is a block diagram of a terminal system including a management station and a plurality of terminals;

Fig. 2 is a simplified block diagram of the man-

agement station shown in Fig. 1;

Fig. 3 is a simplified block diagram of one of the terminals shown in Fig. 1;

Figs 4A and 4B, assembled as shown in Fig. 4, are a flowchart illustrating the operation of the management station during a software distribution operation; and

Fig. 5 is a flowchart illustrating the operation of a terminal during a software distribution operation.

Referring first to Fig. 1, there is shown a terminal system 10 including a management station 12 connected over a communications network 14 to a plurality of terminals 16, referenced individually as terminals 16-1, 16-2, 16-3 up to 16-N. In the preferred embodiment, the terminals 16 are SSTs (self-service terminals) such as ATMs (automated teller machines), adapted for operation by members of the public. Also in the preferred embodiment, the communications network 14 is a public data network to which the management station 12 and the terminals 16 are connected. The management station 12 may be a computer such as a personal computer (PC).

The management station 12 and terminals 16 transfer information in packet mode; that is, information is transferred by means of packets which pass through the communications network 14 via interfaces conforming to the CCITT X.25 standard, for example. However, other types of communications network may be utilized. It should be understood that the system 10 is a peer-to-peer system, that is, any of the terminals 16 can communicate directly with any other terminal 16, as well as with the management station 12. The number of terminals 16 in the system 10 may be quite large, for example there may be several hundred, or over one thousand terminals 16 in the system 10.

Referring now to Fig. 2, there is shown a simplified block diagram of the management station 12. The management station 12 includes a bus 20 to which are connected a processor 22, a display 24, a keyboard 26 and a memory device 28. Also connected to the bus 20 are a control software storage unit 30, which may be a part of the memory 28, and a database unit 32, the function of which will be described hereinafter. The bus 20 is also connected to a transceiver 34 which communicates with the communications network 14 over a channel 36.

Referring now to Fig. 3, there is shown a simplified block diagram of a terminal 16. The terminal 16 includes a bus 50 to which are connected a processor 52, a display 54, a keyboard 56 and a teller control unit 58 which controls the teller functions of the terminal 16, such as cash dispensing, cash receiving, document receiving and processing and the like. Also connected to the bus 50 are a memory 60 and a control software store 62, which may be physically part of the memory 60. The bus 50 is also connected to a

transceiver 64 which communicates with the communications network 14 over a channel 66.

The present invention is concerned with the distribution of a common software package from the management station 12 to the terminals 16. In this connection it will be appreciated that it is desirable that such distribution should be effected with minimum delay, having regard to the delays inherent in the communications network 14, which may cover a wide geographical area, and the possibly large number of terminals 16 destined to receive the common software package.

The operation of the management station 12 during a software distribution procedure will now be described. It should first be understood that the database 32 (Fig. 2) stores a list of all the terminals 16 together with an indication of whether or not they have received the software package to be distributed. It should also be understood that the software package is initially stored in the memory 60 and is transmitted in conventional manner as a series of packets via the transceiver 34 over the communications network 14. Preferably, the management station 12 has the capability of transmitting simultaneously to a plurality of the terminals 16.

Referring now to Figs. 4A and 4B there is shown a flowchart 80 illustrating the operation of the management station 12 during a software distribution procedure. The procedure begins at the start block 82 and proceeds to block 84 where the database 32 (Fig. 2) is checked to determine whether any terminals 16 listed are without the software package. If no terminals are listed as being without the software package, the procedure terminates (block 86). If there are terminals listed as being without the software, the flowchart proceeds to block 88, where the database 32 (Fig. 2) is checked to ascertain whether any terminals are listed as having received the software package. If no, the flowchart proceeds to block 90 and a RECEIVE command is sent to a terminal 16 which does not have the software. It should be noted at this point that if the management station 12 has the necessary capability, then the RECEIVE command may be sent simultaneously to a plurality of the terminals 16. The RECEIVE command has the format

RECV (Filename)

where "Filename" identifies the software package to be distributed. The RECEIVE command conditions the terminal or terminals which receive it to receive the software package identified by its Filename. Next, (block 92) a SEND command effective in the management station 12, having the format

SEND (Filename)

causes the designated software package to be sent

to the terminal (or terminals) which received the RECEIVE command. Next, the database 32 is updated to record the terminal or terminals which have just received the software package, and the flowchart returns over the line 96 to block 84.

If at block 88 it is found that there are terminals listed in the database 32 as having received the software package, then the flowchart proceeds to block 98, and a TRANSFER command is sent to one or more of the terminal 16. The TRANSFER command has the format

XFR (Filename, Destination)

where "Filename" identifies the software package and "Destination" identifies a terminal which is to receive the software package from a terminal which has received the TRANSFER command.

It should be understood that in block 98, a TRANSFER command maybe sent concurrently to a plurality of terminals 16 dependent on the capability of the management station 12. Also, of course, the TRANSFER command will only be sent to as many terminals having the software package as are needed in accordance with the number of terminals which have not yet received the software package.

The flowchart then proceeds to block 100 where it is seen that the management station 12 waits until it has received TRANSFER COMPLETE messages from the terminals having the software package which were designated to send the software package to other terminals. When this operation is completed the flowchart proceeds to block 94 where the database 32 is updated by marking the terminals which have now received the software, whereafter the flowchart returns to block 84 as shown by the line 96.

Referring now to Fig. 5, there is shown a flowchart 120 of the operation of one of the terminals 16 which is to receive a TRANSFER command from the management station 12. Such a terminal 16 can be regarded as an agent of the management station 12 for the transmission of the software package to another one of the terminals 16. The flowchart 120 commences at start block 122 and proceeds to block 124 where it is seen that the terminal receives a TRANSFER command (discussed hereinabove) from the management station 12. Next, as shown by block 126, a RECEIVE command is sent to the terminal specified in the TRANSFER command which is to receive the software package. The software package is then transferred by the agent terminal to the specified terminal (block 128). The agent terminal then sends a TRANSFER COMPLETE message to the management station 12 to advise the management station that the software package has been transmitted to the specified terminal. This terminates the agent terminal's software package transfer operation as shown at block 132.

It will be appreciated that the use of the procedure described hereinabove, wherein terminals which have received the software packages can be controlled to further distribute the software package to other terminals, enables a considerable reduction in the overall time for distributing the software package to other terminals as compared with a procedure wherein the management station itself sends the software package directly to all the terminals.

For example, assume that the management station can perform ten concurrent transmissions, and has to distribute a software package to 1000 terminals ( $T_0$  to  $T_{999}$ ). Assume also that the transmission time for one software package is one hour. Using the described procedure, the management station 12 will send the software package to the terminals  $T_0$  to  $T_9$  during the first hour. At the start of the second hour the management station will command terminals  $T_0$  to  $T_9$  to distribute to terminals  $T_{10}$  to  $T_{19}$  respectively. During the second hour terminals  $T_{10}$  to  $T_{19}$  will receive the software package from their peer terminals  $T_0$  to  $T_9$ . During this time the management station 12 is free to distribute the software to another ten terminals ( $T_{20}$  to  $T_{29}$ ). At the end of the second hour a total of 30 terminals ( $T_0$  to  $T_{29}$ ) will have received the software.

At the start of the third hour the management station 12 will command the terminals that have received the software ( $T_0$  to  $T_{29}$ ) to distribute to terminals  $T_{30}$  to  $T_{59}$  respectively. During the third hour terminals  $T_{30}$  to  $T_{59}$  will receive the software package from their peer terminals  $T_0$  to  $T_{29}$ . During this time the management station is free to distribute the software to another ten terminals  $T_{60}$  to  $T_{69}$ . At the end of the third hour a total of 70 terminals ( $T_0$  to  $T_{69}$ ) will have received the software. The process is repeated at the end of every hour. This results in the number of terminals receiving the software being doubled every hour (i.e. 10, 20, 40, 80, 160, ... etc.) Using this method it will take only seven hours to distribute to 1000 terminals, as compared with the 100 hours it would take for the management station 12 to distribute the software packages directly to the terminals, ten at a time (since  $1000/10 = 100$ ).

#### Claims

1. A method of distributing a software package from a source station (12) to a plurality of terminals (16), characterized by the steps of: transmitting said software package from said source station (12) to at least one of said terminals (16); and utilizing at least one of said terminals (16) which has received said software package to transmit said software package to another of said terminals (16).

2. A method according to claim 1, characterized in that said utilizing step includes the steps of transmitting a transfer command from said source station (12) to one of said terminals (16) which has received said software package, said transfer command specifying another terminal (16) which has not received said software package; and transmitting said software package from said one of said terminals (16) to said another terminal (16).

3. A method according to claim 2, characterized by the steps of transmitting from said one of said terminals (16) which has received said package to said another terminal (16) which has not received said software package a receive command, thereby conditioning said another terminal (16) to receive said software package.

4. A method according to claim 3, characterized by the steps of maintaining at said source station (12) a list of said terminals (16), identifying which terminals (16) have and which terminals (16) have not received said software package.

5. A method according to claim 4, characterized by the steps of: transmitting to said source station (16) a transfer complete message from a terminal (16) which has completed transmission of said software package to another terminal (16), and updating said list of terminals (16).

6. A method according to any one of the preceding claims, characterized in that said step of transmitting said software package from said source station (12) includes transmitting said software package concurrently to a plurality of said terminals (16).

7. A terminal system including a management station (12), a plurality of terminals (16) and connection means (14) adapted to interconnect said management station (12) and said terminals (16), characterized in that said management station (12) is adapted to transmit a software package to at least one of said terminals (16), and is that said management station (12) is adapted to cause a terminal (16) which has received said software package to transmit said software package to another of said terminals (16).

8. A terminal network according to claim 7, characterized in that said management station includes a database (32) adapted to contain a list of said terminals (16) identifying which terminals (16) have and which terminals (16) have not, received said software package.

9. A terminal system according to claim 7 or claim 8, characterized in that said terminals (16) are self-service terminals.

10. A terminal system according to claim 9, characterized in that said self-service terminals (16) are automated teller machines.

5

10

15

20

25

30

35

40

45

50

55

5

**FIG. 1**

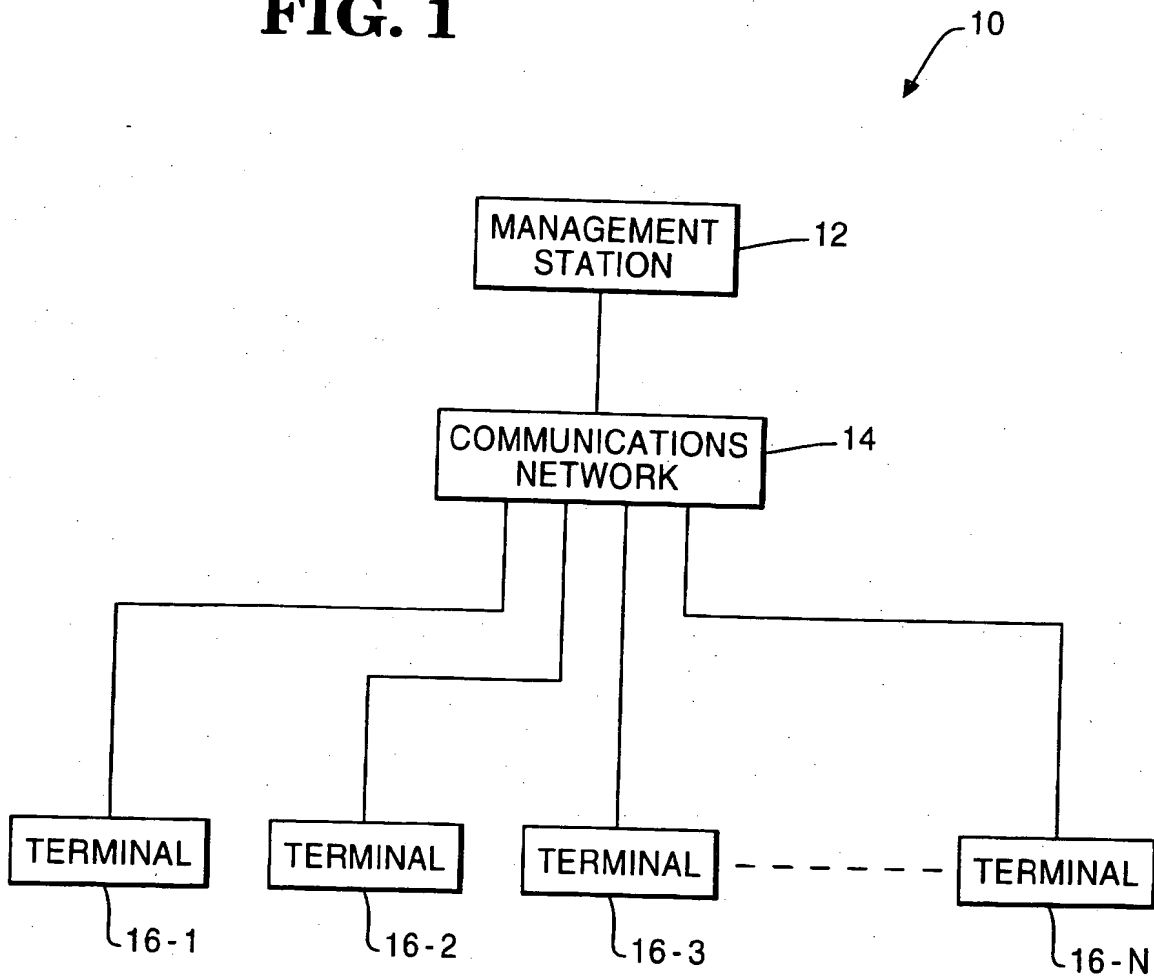


FIG. 2

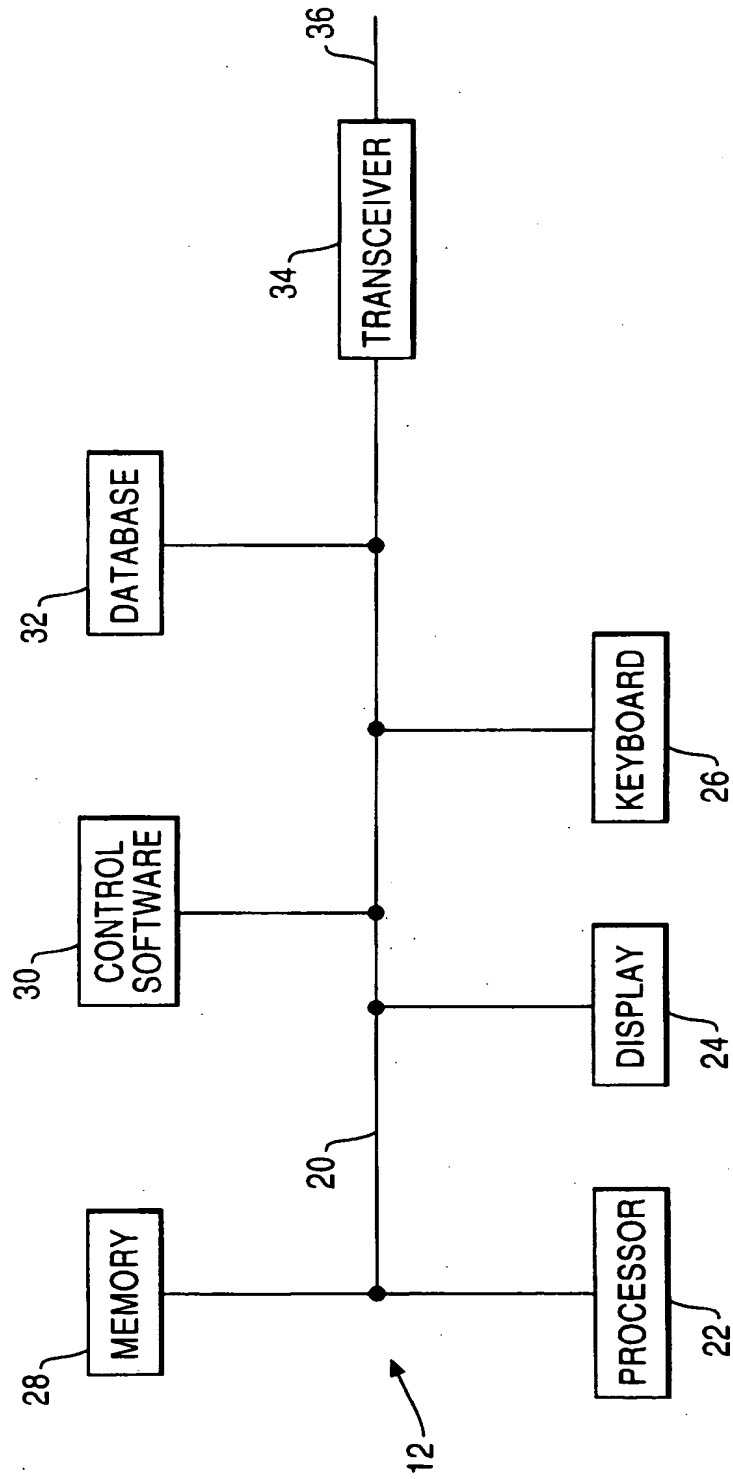


FIG. 3

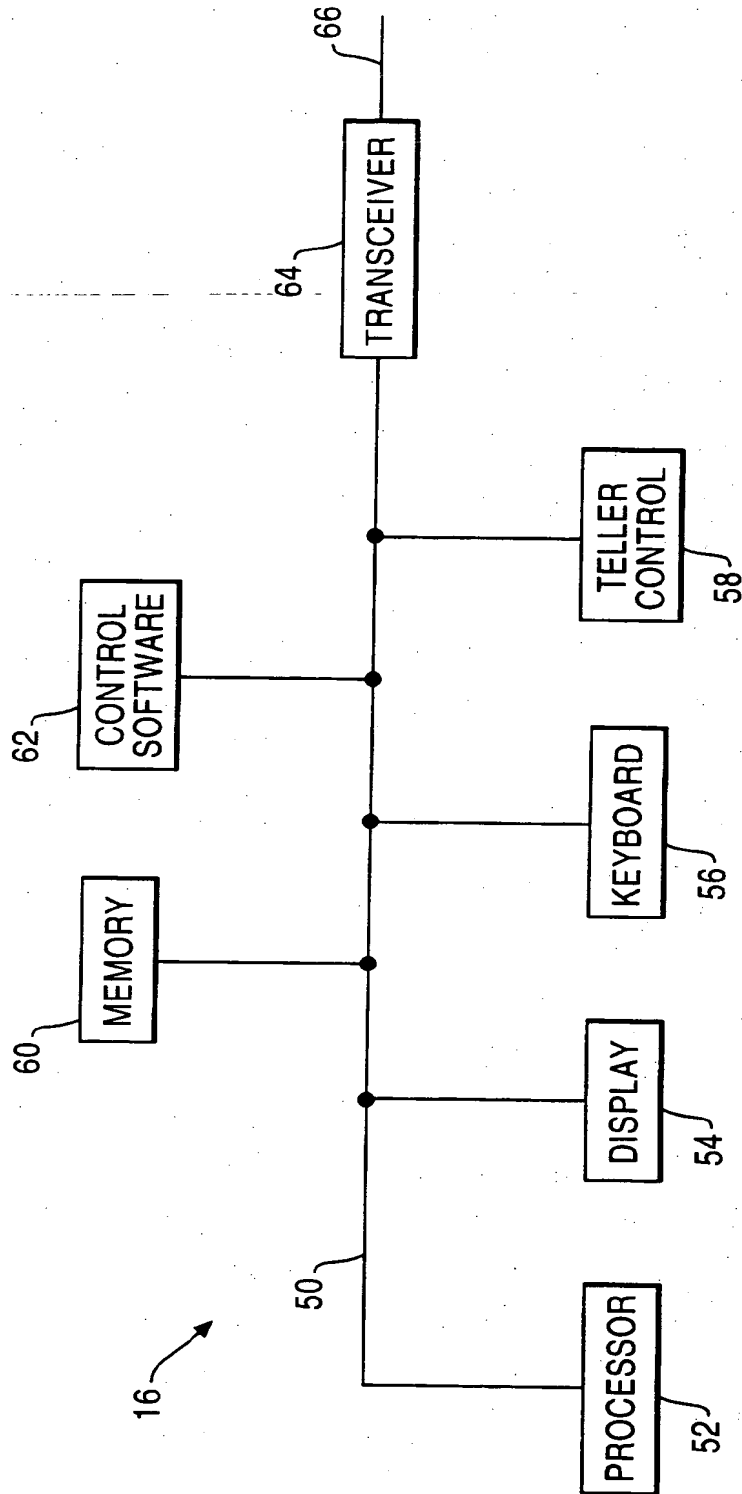
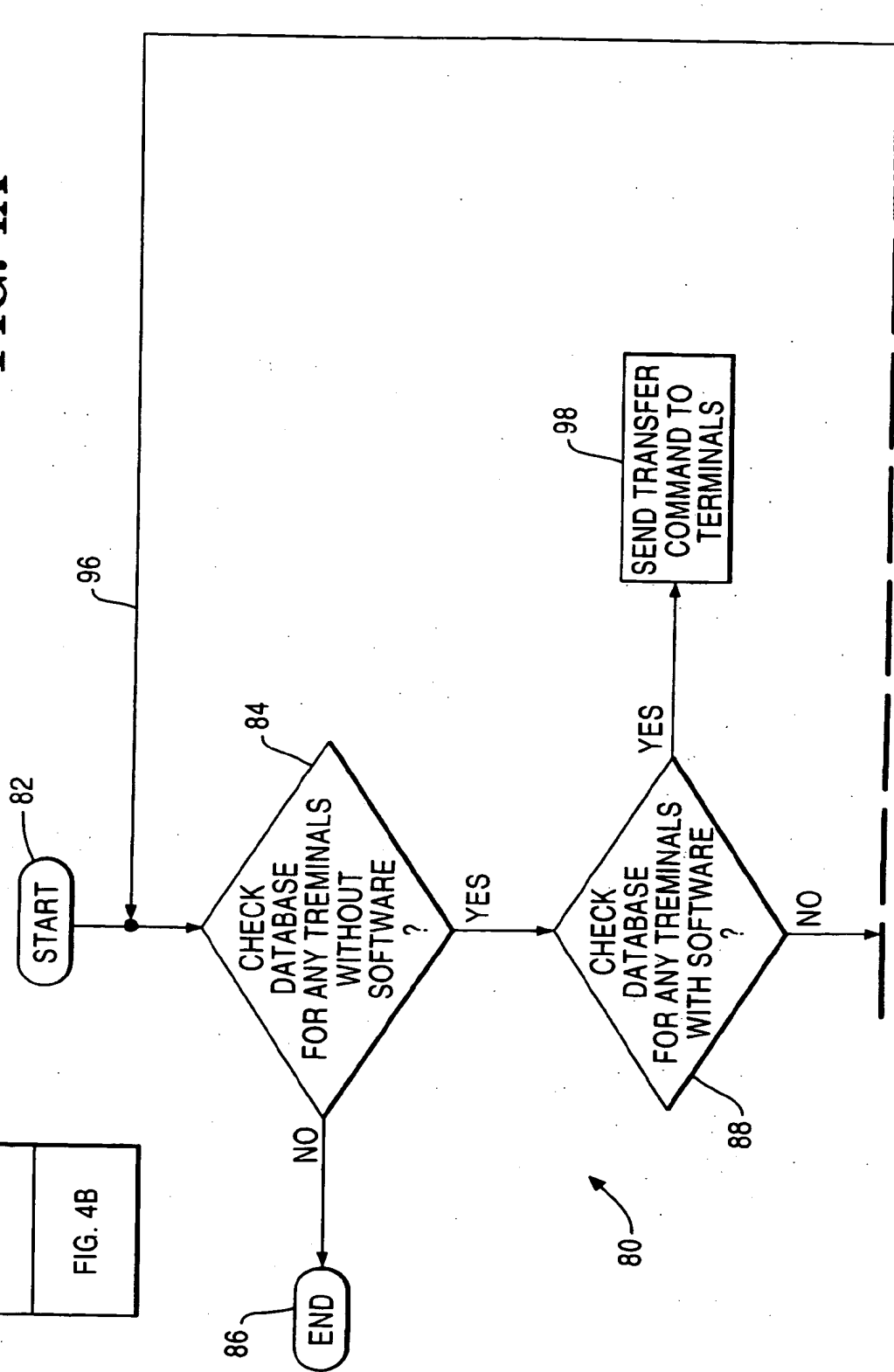




FIG. 4

FIG. 4A
FIG. 4B

FIG. 4A



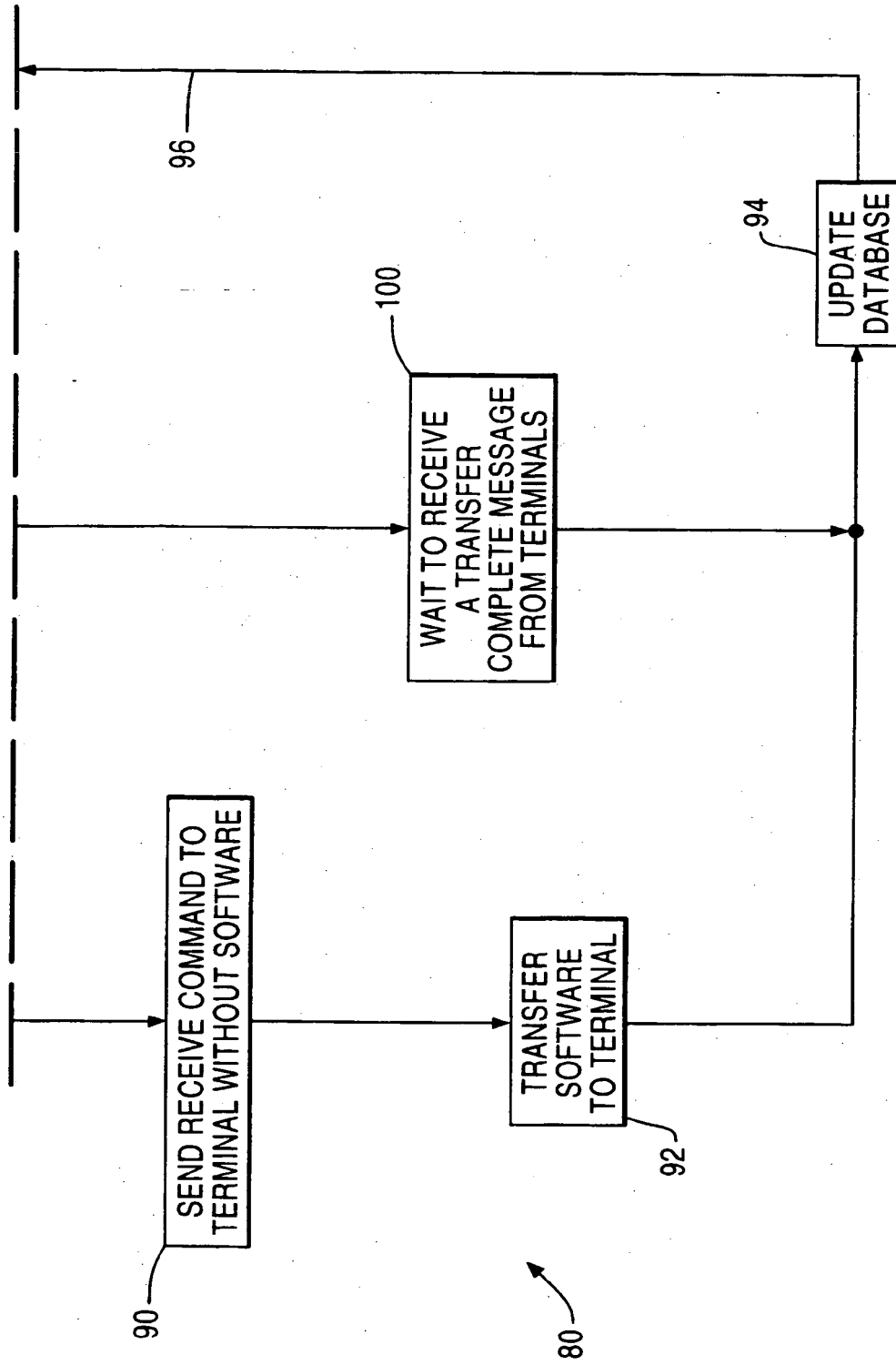
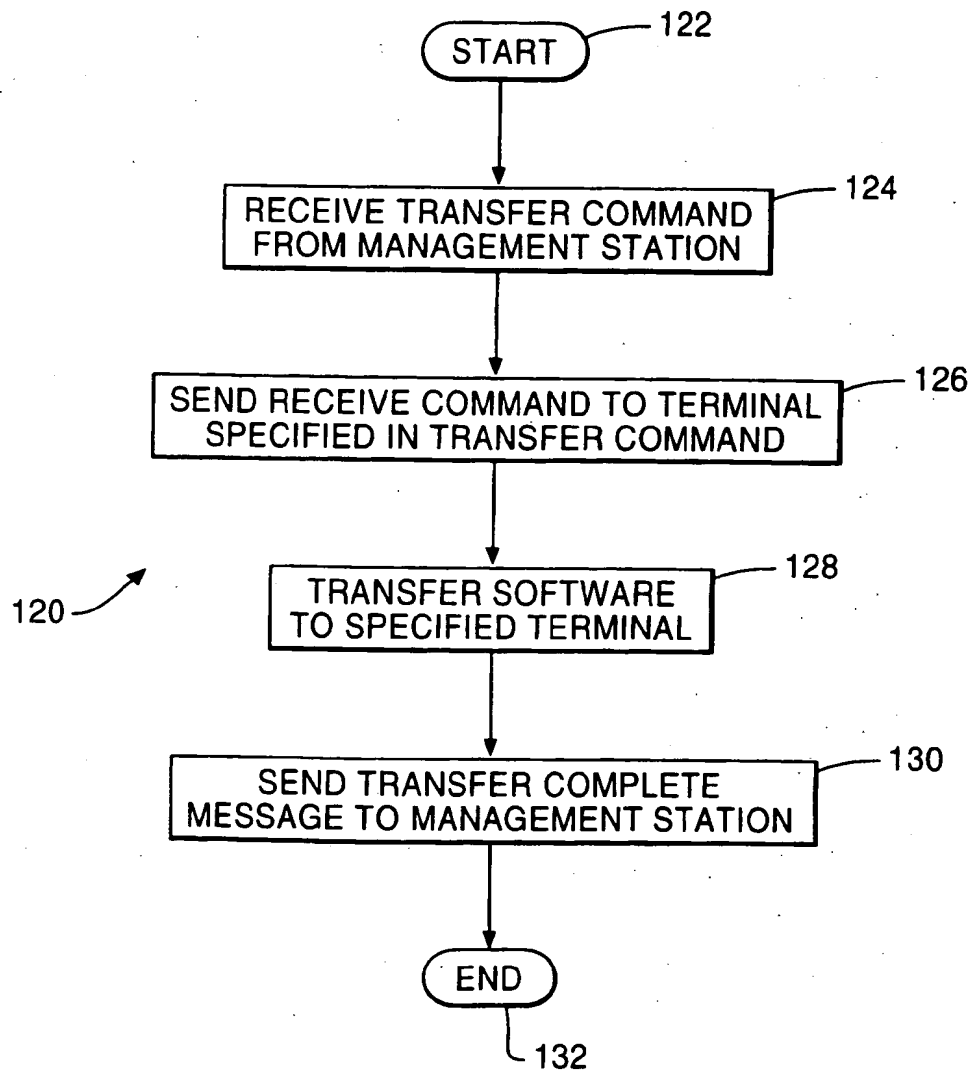


FIG. 4B

**FIG. 5**



**THIS PAGE BLANK (USPTO)**

FEB 26 2001

SHOOK, HARDY & BACON  
Attn. KIRCHER, W.  
One Kansas City Place  
1200 Main Street  
Kansas City, Missouri 64105-2118  
UNITED STATES OF AMERICA

Date: 21/02/2001

**THIS PAGE BLANK (USPTO)**